

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería de Servicios y Tecnología de  
Telecomunicaciones (EUR-ACE®)**

**TRABAJO FIN DE GRADO**

**Comunicador Android sobre Reloj Inteligente**

**Ignacio de Diego de Somonte y Cañestro  
Tutor: Eduardo Boemo**

**Junio 2018**



# **Comunicador Android sobre Reloj Inteligente**

**AUTOR: Ignacio de Diego de Somonte y Cañestro**

**TUTOR: Eduardo Iván Boemo Scalvinoni**



**Laboratorio de Sistemas Digitales  
Dpto. Tecnología Electrónica y de Comunicaciones  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Junio de 2018**



## **Resumen (castellano)**

Este Trabajo Fin de Grado tienen como objetivo el desarrollo de una aplicación para relojes inteligentes, que sirve como comunicador para personas con TEA (Trastorno del Espectro Autista).

Forma parte de un proyecto mayor del DsLab con colaboración de profesionales de educación especial del CEIP Romero Peña de la Solana (Ciudad Real), que pretende realizar una familia de cuatro comunicadores: uno para tableta, otro para reloj, otro con una extensión para controlar aparatos eléctricos (ventilador, alarmas, etc.), y finalmente, una versión de tableta, manejada por la cabeza.

Este comunicador permite a través de un menú principal y diferentes menús secundarios con pictogramas personificados, que una persona con algún tipo de discapacidad o problema de lenguaje se exprese mediante los pictogramas con su familia, profesores, etc. Hasta hace pocos años, esta herramienta era una carpeta de anillas con hojas.

Como plataforma SW/HW se ha elegido Wear Os y un reloj Motorola modelo Moto 360. La aplicación, cuenta además con un sistema de analíticas que servirá para que los especialistas en educación especial puedan monitorizar datos y modos de uso, que sean la base para futuras versiones.

Para concluir, la aplicación, Quiero decirte, estará disponible tanto en Google Play como en la página oficial del proyecto y se podrá descargar de manera gratuita una vez que los profesionales de educación de diversos colegios la prueben y se realice un segundo ciclo de diseño.

## **Palabras clave (castellano)**

Android, Android Wear, Wear Os, reloj inteligente, aplicación, Trastorno del Espectro Autista, educación, pictogramas.



## **Abstract (English)**

This Bachelor Thesis aims to develop an application for smartwatches, which serves as a communicator for people with ASD (Autism Spectrum Disorder).

It is part of a larger DsLab project with the collaboration of special education professionals from CEIP Romero Pena de la Solana (Ciudad Real), which aims to create a family of four communicators: one for a tablet, another for a clock, another with an extension to control electrical appliances (fan, alarms, etc), and finally, a tablet version, operated by the head.

This communicator allows through a main menu and different secondary menus with personified pictograms, that a person with some type of disability or language problem, express through the pictograms with his family, teachers, etc. Until a few years ago, this tool was a ring binder with leaves.

As a SW / HW platform, Wear Os and a Motorola model Moto 360 watch have been chosen. The application also has an analytical system that will allow special education specialists to monitor data and modes of use, which are the basis for future versions.

To conclude, the application, Quiero decirte, will be available on Google Play and on the official website of the project and can be downloaded for free once the education professionals from various schools try it out and a second design cycle is carried out.

## **Keywords (inglés)**

Android, Android Wear, Wear Os, smartwatch, application, Autism Spectrum Disorder, education, pictograms.





## ***Agradecimientos***

*En un primer lugar, me gustaría agradecer a Eduardo Boemo, mi tutor, por todo el apoyo que a lo largo del curso y las facilidades que ha proporcionado para realizar este Trabajo de Fin de Grado.*

*También me gustaría agradecer a mis amigos y compañeros de clase, por los momentos que hemos vivido juntos.*

*Y por último, a mis padres y hermanos, que han estado en todos los momentos, ya sean buenos o malos, mostrando su apoyo.*



# ÍNDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
1.3	Organización de la memoria.....	3
2	Estado del arte .....	5
2.1	DictaPicto .....	5
2.2	Pictogramas.es .....	5
2.3	EmoPlay.....	6
2.4	Grace app.....	7
2.5	Resumen .....	7
3	Diseño.....	9
3.1	Finalidad .....	9
3.2	Entorno de programación .....	9
3.2.1	Sistema operativo .....	9
3.2.2	Wear OS (Android Wear).....	11
3.2.3	Android Studio .....	12
3.2.4	Idioma de la aplicación.....	12
3.3	Especificaciones funcionales .....	13
3.3.1	Indicaciones generales.....	13
3.3.2	Pautas generales de diseño .....	13
3.3.3	Configuración/Ajustes .....	14
3.3.1	Registros de usos .....	15
4	Desarrollo .....	17
4.1	Primeros pasos.....	17
4.1.1	Búsqueda y recogida de información .....	17
4.1.2	Herramientas utilizadas .....	17
4.2	Aplicación.....	18
4.2.1	Nombre y duración del proyecto .....	18
4.2.2	Estructura del proyecto en Android Studio .....	18
4.2.2.1	Carpeta Manifest.....	18
4.2.2.2	Carpeta Java.....	19
4.2.2.3	Carpeta res .....	20
4.2.2.4	Gradle.scripts.....	20
4.2.3	Estructura de la Aplicación.....	21
4.2.3.1	Menú Principal de la aplicación .....	21
4.2.3.1	Menú Alimentos .....	21
4.2.3.2	Menú Personas.....	22
4.2.3.1	Menú Objetos .....	22
4.2.3.2	Menú Lugares .....	23
4.2.3.3	Menú Aseo.....	23
4.2.3.4	Menú Interacción.....	23
4.2.3.5	Quiero/Ayuda/No/Borrar .....	24
4.2.3.6	Menú Ajustes .....	24
4.3	Estructura final aplicación .....	25
4.4	Tiempo por pantalla.....	26
4.5	Principales problemas resueltos.....	27
5	Integración, pruebas y resultados .....	29
5.1	Publicación .....	29
5.2	Resultados.....	29

6 Conclusiones y trabajo futuro.....	31
6.1 Conclusiones.....	31
6.2 Trabajo futuro .....	31
Referencias .....	33
Glosario .....	35
Anexos .....	I
A      Conexión del smartwatch con Android Studio.....	I
B      Publicación en Google Play.....	V
C      Referencias Imágenes .....	IX

## ÍNDICE DE FIGURAS

2.1 DICTAPICTO .....	5
2.2 PICTOGRAMAS.ES .....	6
2.3 EMOPLAY .....	6
2.4 GRACE.....	7
3.1 LOGO WEAR OS .....	11
3.2 ANDROID STUDIO .....	12
3.3 OBJETOS .....	14
3.4 LUGARES .....	14
3.5 ALIMENTOS .....	14
3.6 NO .....	14
3.7 AYUDA .....	14
3.8 INTERACCIÓN .....	14
3.9 ASEO.....	14
4.1 MANIFEST .....	18
4.2 SALIR APP .....	19
4.3 SALIR APP .....	19
4.4 ARCHIVOS JAVA .....	19
4.5 CARPETA RES.....	20
4.6 MENÚ PRINCIPAL.....	21

4.7 PANTALLA ALIMENTOS .....	22
4.8 PANTALLA PERSONAS .....	22
4.9 PANTALLA OBJETOS .....	22
4.10 PANTALLA LUGARES .....	23
4.11 PANTALLA ASEO .....	23
4.12 PANTALLA INTERACCIONES.....	23
4.13 PANTALLA PIN.....	24
4.14 PIN INCORRECTO .....	24
4.15 PIN CORRECTO .....	25
4.16 PANTALLA CONFIGURACIÓN .....	25
4.17 DIAGRAMA APLICACIÓN.....	26
4.18 EMULADOR.....	27
4.19 TAMAÑO MIPMAP .....	28

## ÍNDICE DE TABLAS

TABLA 3.1 COMPARATIVA WATCHOS VS WEAR OS .....	10
TABLA 3.2 NIVEL DE VENTA PRINCIPALES COMPAÑÍAS .....	10



# 1 Introducción

---

## 1.1 Motivación

Los expertos coinciden en que el uso de las tecnologías (ordenadores, dispositivos móviles, tabletas, relojes inteligentes) utilizados de la manera adecuada, pueden resultar muy beneficiosos para personas con problemas en la comunicación. Por lo tanto, una motivación de este TFG es ayudar a personal a través de la tecnología.

En la actualidad hay multitud de aplicaciones que tratan temas relacionados. Los expertos aconsejan a las familias que se usen solo como apoyo a otras terapias y metodologías, en ningún caso se debería utilizar esta aplicación como único recurso para tratar los problemas de comunicación.

Este proyecto surge por la necesidad de corregir deficiencias que se habían encontrado en otras aplicaciones de las mismas características en dispositivos móviles o tabletas y llevarlo a un nuevo tipo de dispositivo aún más pequeño y que se puede llevar siempre encima como son los smartwatches.

Las ventajas respecto a otras aplicaciones similares que podríamos destacar son:

- Sencillez.
- Pictogramas estándares (imágenes) ya reconocidos por los usuarios.
- Minimización del nivel de coordinación visomotor y precisión necesaria para su uso.
- Reducción de la estimulación visual en lo referente a colores, letras y pictogramas.
- Inicialización automática.
- Pocos elementos por pantalla.

Este proyecto se encuadra en una acción del DsLab financiada por la Fundación Vodafone, cuyo objetivo es mejorar la calidad de vida y de comunicación de personas con discapacidades. Los cuatro proyectos que lo componen son:

- Comunicador en tableta inteligente.
- Comunicador en tableta inteligente con movimiento de cabeza.
- Reloj inteligente/Smartwatch.
- Comunicador con módulo de control de aparatos eléctricos.

Por tanto la comunicación entre estos proyectos y la colaboración ha sido fundamental para conseguir el objetivo principal.

Durante el desarrollo se han realizado diferentes cursos, para poder cumplir con las especificaciones dadas en la primera reunión. A medida que se ha ido avanzado y se han celebrado diferentes reuniones para revisar los contenidos, actualizándolos o modificándolos si eran necesarios, se ha conseguido desarrollar una primera versión de la aplicación.

## **1.2 Objetivos**

El objetivo de este proyecto es crear una aplicación para smartwatch (en español: reloj inteligente) con sistema operativo Android Wear que sirva como comunicador. El objetivo primordial es la ayuda en la comunicación tanto en niños como en adultos con Trastorno del Espectro Autista (TEA).

Con este sistema desarrollado se pretende que personas con niveles cognitivos muy bajos puedan ser asistidas en la comunicación. La aplicación para Android Wear cuenta con un menú gráfico principal y varios menús gráficos secundarios, todos ellos con pictogramas proporcionados por la asociación ARASAAC para poder así facilitar la transición de la información, gracias a estos pictogramas habituales para los usuarios, sin necesidad de usar la voz.

Otro de los objetivos es que a través de la herramienta de Firebase/Google Analytics se monitorice la actividad de los usuarios. Gracias a ésta se conseguirá optimizar el rediseño permanente de la aplicación a través del conocimiento de su uso (Big-data), pudiendo así mejorar la aplicación hasta conseguir el nivel de estabilidad y practicidad más adecuado para los usuarios.

El resultado de este proyecto es una aplicación con un sistema de monitorización integrado para ayudar a las personas con TEA. A muchas de estas personas les resulta muy complicado las relaciones sociales, la comunicación y pensar en abstracto. Este proyecto trata de que la comunicación resulte un poco menos complicada. En todo momento se enfatiza la sencillez de los pictogramas, el uso intuitivo, y el fácil acceso.

La aplicación tratará de ayudar a cubrir necesidades importantes para estas personas como:

- Fortalecimiento de la autonomía y el desarrollo de las capacidades de autocontrol.
- Simplicidad de uso a modo de mitigar frustración, ofreciendo un refuerzo alto e inmediato.
- Estimulación visual que proporciona una gran motivación y atención por parte de los usuarios.
- Adaptable a las características de usuarios.



### ***1.3 Organización de la memoria***

La memoria consta de los siguientes capítulos:

- Capítulo 1: Introducción.
- Capítulo 2: Estado del Arte.
- Capítulo 3: Diseño.
- Capítulo 4: Desarrollo.
- Capítulo 5: Integración, pruebas y resultado.
- Capítulo 6: Conclusiones y trabajo futuro.

Para facilitar una lectura fluida, a lo largo de la Memoria se ha decidido utilizar los términos técnicos relacionados con Android, en el idioma de origen, inglés, y sin ser distinguidos con Italic o comillas.



## 2 Estado del arte

---

Las tiendas de aplicaciones para Wear Os están muy limitadas y contienen muy pocas aplicaciones si se compara con las de Android. Aunque poco a poco, gracias a las grandes compañías, hay cada vez más aplicaciones. Estas aplicaciones suelen ser de personificación, herramientas (calculadora, calendario, notas,...), bancos, juegos mientras que otras aplicaciones que en Android son bastante útiles, no consiguen que se desarrollen para este sistema operativo.

### 2.1 DictaPicto

Aplicación disponible gratuitamente para dispositivos con sistemas operativos Android. Es una aplicación creada para trabajar de forma educativa o terapéutica.

Esta aplicación es un traductor de voz o texto a pictogramas. Funciona al revés de lo esperado, la comunicación que se busca es la del acompañante con el usuario. Por lo que el acompañante tendrá que escribir o grabar un mensaje de voz en la aplicación. Esta identificará los conceptos más importantes y buscará el pictograma que más se adecue.



#### 2.1 DictaPicto<sup>i</sup>

Cuenta con una serie de limitaciones, ya que la comunicación no puede ser del usuario a otra persona y solo tiene versión para Android, no para Wear Os.

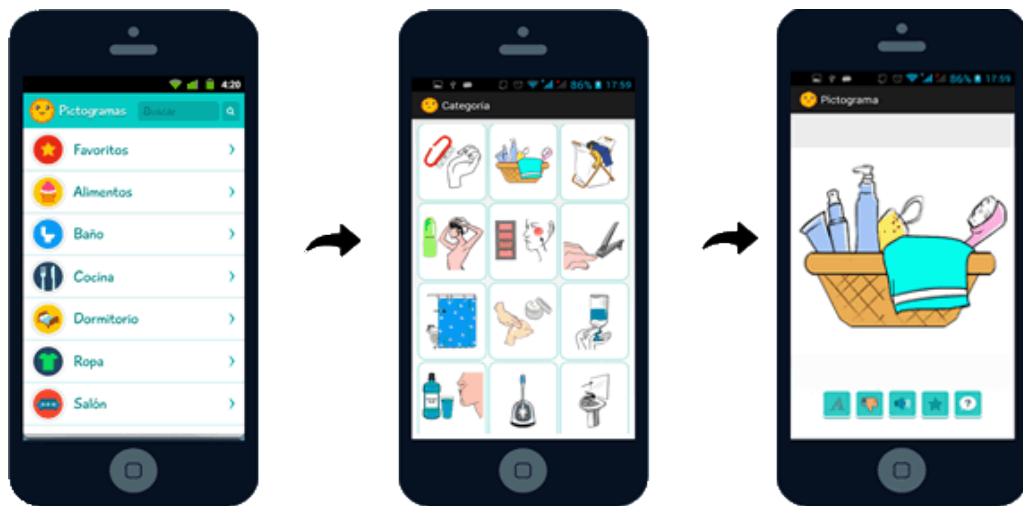
### 2.2 Pictogramas.es

Es una aplicación que se encuentra para dispositivos Android. Actualmente se está desarrollando para iOS.

Cuenta con una serie de pictogramas y de locuciones de múltiples categorías. Una vez que la aplicación esté descargada en el dispositivo, no es necesario la conexión a internet, lo que hace a esta aplicación muy interesante.

Recomendada para familias, colegios, usuarios con TEA o Síndrome de Down, o simplemente personas que necesitan fortalecer la comunicación oral.

Una de las ventajas más importantes es que se puede buscar los pictogramas introduciendo el nombre o buscando la imagen directamente. Por lo que dependiendo del nivel se puede usar una cosa u otra.



2.2 Pictogramas.es<sup>ii</sup>

### 2.3 EmoPlay

Disponible gratuitamente para sistemas operativos Android, Windows, iOS. Entrena el reconocimiento de las expresiones faciales.

Mediante una elección previa de la emoción que se quiere trabajar y una explicación, el usuario tendrá que hacer los diferentes gestos que caracterizan esa emoción a la cámara del dispositivo. La aplicación evaluará automáticamente si la expresión es adecuada o no. Con esto se pretende que los usuarios comprendan y se familiaricen con diferentes expresiones faciales y que en un futuro puedan hacer ellos mismos todas las expresiones.



2.3 EmoPlay<sup>iii</sup>

Esta aplicación tampoco está disponible para Wear Os.

## **2.4 Grace app**

Esta aplicación se encuentra solo para dispositivos iOS. Cuenta con una interfaz muy sencilla, diseñada para ayudar a comunicarse de manera afectiva, mediante la construcción a través de los diferentes pictogramas de frases. Puede personalizarse añadiendo nuevos pictogramas y vocabulario.

Comparado con las demás aplicaciones, esta podría estar en un nivel de dificultad mayor, ya que el objetivo es formar frases mientras con las otras con un solo pictograma era suficiente.



**2.4 Grace<sup>iv</sup>**

## **2.5 Resumen**

Aunque hay una cantidad considerable de aplicaciones para personas con TEA, no hay ninguna que sea perfecta o que abarque todos los ámbitos necesarios para la ayuda. También hay que tener en cuenta que por ahora no hay ninguna aplicación en cualquiera de las tiendas de aplicaciones, ya sea para Wear Os o para watchOs, para personas con TEA. Por lo que la aplicación desarrollada en este Trabajo de Fin de Grado podría ser la primera en cubrir estas necesidades.



## 3 Diseño

---

### 3.1 Finalidad

Este proyecto como ya se ha comentado con anterioridad tiene como objetivo facilitar a las personas con TEA la comunicación de acciones de la vida cotidiana con su familia y profesores.

Esta aplicación cuenta con un menú principal con los pictogramas de diferentes categorías, una vez que el usuario entre en una categoría se le mostraran varios pictogramas más específicos, con la finalidad de que el usuario llegue a la acción que necesite sin muchas complicaciones. A su vez cuenta con una función de ajustes creada específicamente para las familias y profesores con él se podrá cambiar el tiempo por defecto de estancia en una pantalla.

Con esta serie de pantallas y ajustes se pretende construir una aplicación lo más personificada posible para los usuarios, logrando así que en un dispositivo como es el reloj inteligente, se puede tener a disposición en todo momento, una herramienta que sirva de comunicador.

Aunque otros dispositivos como son la tableta o un dispositivo móvil pueden ser muchos más útiles dependiendo de la circunstancia, ya que son más personificables, con este pequeño dispositivo se ha conseguido crear un comunicador a medida y muy útil para estas personas.

### 3.2 Entorno de programación

Tras haber hecho un pequeño estudio de mercado entre los distintos sistemas operativos que los smartwatches tienen en la actualidad, comparado la calidad precio de los dispositivos nos centraremos poco a poco en explicar la elección de estos.

#### 3.2.1 Sistema operativo

Actualmente en el mercado de los smartwatches hay dos sistemas que como ocurre en dispositivos móviles y tabletas predominan sobre los demás. Son el caso de watchOs (sistema operativo del Apple Watch) y Wear Os (conocido anteriormente como Android Wear).

Ambos sistemas operativos son bastante parecidos pero se pueden encontrar algunas diferencias. En la siguiente tabla se comparará las principales características de los sistemas operativos.

<b>Controles</b>	La pantalla táctil, "corona digital" depende del fabricante	Pantalla táctil, esfera "Corona digital"
<b>Control de voz</b>	Asistente de Google	Siri
<b>Mercado de aplicaciones</b>	Google Play	Tienda de aplicaciones de Apple
<b>Control de estado físico y salud</b>	Google Fit (o la aplicación de fabricante de relojes)	Apple Health
<b>Sensor de frecuencia cardíaca</b>	Depende del fabricante	Sí
<b>Impermeable</b>	Depende del fabricante	Splash / resistente al agua para IPX7
<b>Compatibilidad con teléfonos inteligentes</b>	Android 4.3 o superior e iOS 9 o superior	iPhone 5 y posterior
<b>Precio</b>	\$ 100 +	\$ 250 +

**Tabla 3.1 Comparativa watchOs vs Wear Os<sup>v</sup>**

Una vez que hemos comparado las características fundamentales nos centraremos en el precio. El sistema operativo Wear Os por su parte cuenta con dispositivos de prestigiosas marcas como Motorola, LG, Samsung y muchas más marcas con precio muy diferentes. Se pueden encontrar smartwatches con un precio de salida superior a los 100 \$. En cambio, los dispositivos con watchOs cuentan con gran exclusividad, ya que la empresa multinacional estadounidense Apple Inc. es la propietaria de este sistema operativo. WatchOs es utilizada en los smartwatches fabricados por la propia empresa como ocurre en el mercado de los dispositivos móviles con el sistema operativo iOS y el iPhone. Por esta exclusividad, el precio de los smartwatches fabricados por esta empresa asciende a un precio de salida superior a los 250\$.

Actualmente todos estos dispositivos han ido perdiendo fuerza con el transcurso del tiempo, produciéndose así un estancamiento en las ventas. En la siguiente tabla se podrá comparar el volumen de ventas de las cinco compañías con los niveles de venta más alto.

Top 5 Wearable Companies by Shipment Volume, Market Share, and Year-Over-Year Growth, Q4 2017 (shipments in millions)					
Company	4Q17 Shipment Volumes	4Q17 Market Share	4Q16 Shipment Volumes	4Q16 Market Share	Year Over Year Change
Apple	8.0	21.0%	5.1	14.4%	57.5%
Fitbit	5.4	14.2%	6.5	18.5%	-17.3%
Xiaomi	4.9	13.0%	5.2	14.7%	-4.5%
Garmin	2.5	6.5%	2.3	6.6%	4.7%
Huawei	1.6	4.3%	0.8	2.4%	93.2%
Others	15.6	41.0%	15.3	43.5%	1.7%
Total	37.9	100.0%	35.2	100.0%	7.7%
Source: IDC Worldwide Quarterly Wearables Tracker, March 1, 2018					

**Tabla 3.2 Nivel de Venta Principales Compañías<sup>vi</sup>**

En la tabla se puede ver claramente que la compañía Apple con su Apple Watch es la compañía con niveles de venta mayor, cuenta con 21% de las ventas del mercado de relojes



inteligentes superando en más de 6.5% a sus competidor más cercano. Pero si a la hora de comparar solo se tiene en cuenta los sistemas operativos y no las compañías que los fabrican, Wear Os se pondría por delante de watchOs (Apple). Esto se debe a que hay una cantidad superior de fabricantes que utilizan Wear Os como sistema operativo y la exclusividad de watchOs que solo se encuentra en dispositivos producidos por Apple.

Tras finalizar estas comparativas se decide utilizar el sistema operativo Wear Os para desarrollar la aplicación. Ya que se pueden encontrar dispositivos a precios más asequibles y hay una cantidad mayor de personas que utilizan este sistema operativo que es lo que al final a cualquier aplicación le interesa.

### 3.2.2 Wear OS (Android Wear)

Wear OS o como anteriormente era conocido Android Wear es un sistema operativo tanto para relojes inteligentes como para pulseras inteligentes que está basado en el sistema Android, perteneciente a la compañía Google. Este sistema operativo fue presentado el 18 de marzo de 2014.



3.1 Logo Wear OS<sup>vii</sup>

Aunque en un primer lugar Google no quiso que los fabricantes tuvieran la opción de incorporar su propia personalización en los dispositivos con el sistema operativo, finalmente si se permitió que los fabricantes y desarrolladores realizaran las personalizaciones que consideran oportunas. Siendo así mucho más atractivo para las compañías el desarrollo de dispositivos con el sistema Wear Os.

Hasta la fecha se han ido modificando y actualizando el sistema operativo, creando nuevas funciones, útiles para los desarrolladores y para los usuarios, y se han dejado obsoletas otras funciones que tenían un uso mínimo o que se han sido incorporadas dentro de otras funciones. Las versiones del sistema operativo con algunas de sus principales características son:

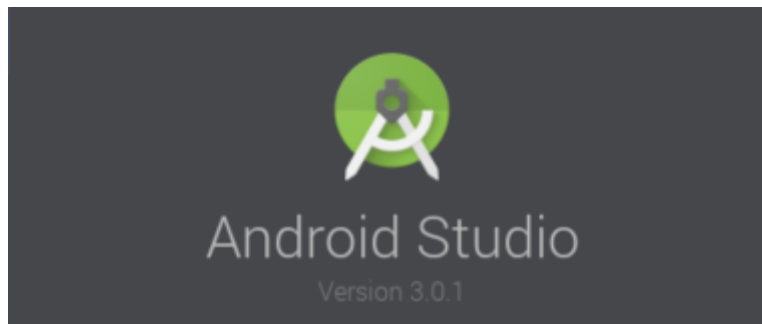
- Android Wear 1.0: primera versión basada en Android 4.4.2 KitKat.
- Android Wear 1.3: se incorpora la conexión WI-FI, así como la duración de la batería y el aumento de velocidad.
- Android Wear 1.4: soporte oficial de altavoces y capacidad de enviar directamente mensajes de voz por medio del micrófono del propio reloj.
- Android Wear 1.5: correcciones de errores.

- Android Wear 2.0: ha sido la actualización más novedosa desde su lanzamiento, cuenta con numerosas novedades. Mejora de la escritura, optimización de las interfaces, la posibilidad de instalar aplicaciones desde el propio reloj usando la aplicación Google Play, son unas de las pocas novedades que traería esta gran actualización.
- Android Wear 2.6: nuevas tarjetas de notificación, notificación con texto adaptable, mejora en la detección del control gestual, la cortina de ajustes muestra ahora el estado de la conexión, accesos directos en las esferas para las aplicaciones recientes.

Como se puede observar el sistema operativo está en un constante proceso de actualización, incorporando cada cierto tiempo pequeñas mejoras para que el uso de los dispositivos sea lo más atractivo posible para los usuarios.

### 3.2.3 Android Studio

Para este proyecto se ha decidido utilizar el software gratuito Android Studio [2] que es el software oficial para implementar aplicaciones Android y Wear Os. Aunque hay otros softwares donde se puede implementar las aplicaciones como puede ser Eclipse, nos hemos decantado por este ya que cuenta con todas las actualizaciones y con un servicio técnico muy superior a los demás softwares no oficiales.



### 3.2 Android Studio

A la hora de elección de este software también se tuvo en cuenta que con Android Studio viene incorporado un emulador de dispositivos con el que se puede ir haciendo las pruebas de la aplicación a medida que se realiza. Aunque en nuestro caso todas las pruebas se realizaron por medio físico con un smartwatch para tener absoluta certeza que se estaba realizando correctamente.

En el [Anexo A](#) se puede ver detalladamente como conectar el smartwatch con Android Studio mediante la depuración USB de un dispositivo móvil y el smartwatch.

### 3.2.4 Idioma de la aplicación

Esta aplicación al estar dirigida para un grupo de personas con unas necesidades especiales, ya que requieren del uso de pictogramas personificados con ciertas palabras, se ha desarrollado en español.

Si por algún motivo se quisiese desarrollar esta aplicación en otro idioma sería una tarea más que sencilla. El desarrollador de aplicaciones solo tendría que cambiar los pictogramas antiguos por los nuevos con las palabras en el idioma correspondiente, cambiar los títulos y la sección de ajustes. Sin necesidad de entender el código elaborado con anterioridad.

### **3.3 Especificaciones funcionales**

En esta sección se explicarán las especificaciones funcionales requeridas a la hora de implementar la aplicación.

#### **3.3.1 Indicaciones generales**

Las indicaciones que las profesoras Juana Domínguez Arroyo y María José García Ariza (profesoras del CEIP Romero Peña de la Solana, donde se harán todas las pruebas de la aplicación) fueron muy específicas desde el primer momento.

Partiremos desde el smartwatch apagado y la aplicación ya instalada. Una vez que el smartwatch se encienda deberá arrancar de forma directa la aplicación sin dar ninguna otra opción al usuario.

Una vez dentro de la aplicación, no se permitirá salir de ella por medio de ningún botón para utilizar otras aplicaciones. Se hace una excepción con los botones físicos ya que no se puede anular las funcionalidades impuestas por los fabricantes.

La primera pantalla aparecerá un menú con las distintas categorías. Para poder visualizar todas categorías el usuario deberá deslizar el dedo. Para seleccionar una categoría y acceder a los distintos pictogramas que la componen se dará un toque en la categoría deseada.

Otro aspecto muy importante es que deberá volver a la pantalla inicial si pasan más de un tiempo definido. Este tiempo deberá poder ser cambiado por los profesores o la familia.

#### **3.3.2 Pautas generales de diseño**

En cuanto al diseño ambas profesoras también fueron muy específicas para facilitar el uso de la aplicación.

Todos los pictogramas tanto los usados como categorías como los que hay dentro de cada una de ellas deberán ser lo más sencillos posibles. Se utilizarán los pictogramas del portal ARSAAC. [1]

Las pantallas no deberán tener demasiadas estimulaciones, es decir, pocos colores y líneas con las que el usuario se pudiera distraer.

Cada categoría es representada por un color diferente que deberá ser respetado. Este borde se deberá ver claramente para que los usuarios lo relacionen directamente con la categoría a la que pertenece

En esta primera versión de la aplicación cada categoría contendrá un número máximo de cinco pictogramas que sean intuitivos y los más realistas posibles.

A continuación, se muestra las ocho categorías con sus respectivos bordes:

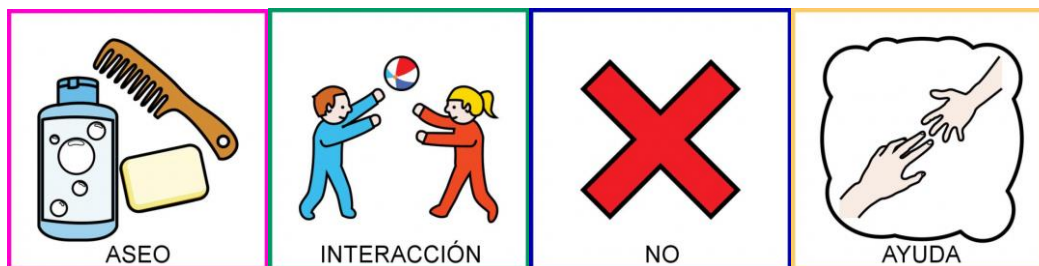


**3-3 Personas**

**3.5 Alimentos**

**3.4 Lugares**

**3.3 Objetos**



**3.9 Aseo**

**3.8 Interacción**

**3.7 Ayuda**

**3.6 No**

En un primer lugar los pictogramas que nos proporciona el portal ARASAAC [1] no tienen los marcos correspondientes, por lo que ha sido tarea del grupo de TFGs retocar los pictogramas e introducir los marcos con su color adecuado.

### 3.3.3 Configuración/Ajustes

La aplicación cuenta con una opción en el menú principal muy importante. Se trata de la configuración de la aplicación.

A esta parte de la aplicación se accederá con una clave de acceso para pasar al modo configuración.

Una vez que se esté en el menú de configuración se podrá cambiar la contraseña que por defecto viene y la opción más importante para las profesoras que es el tiempo de instancia máximo por pantalla sin estar utilizándose, ya comentado con anterioridad.

### **3.3.1 Registros de usos**

Otro parte muy importante son los registros de uso. Se pretende que haya un control de lo que hacen los usuarios y poder acceder de esos datos desde un ordenador.

Los registros que se deberán poder acceder desde el ordenador son:

- Entradas por mes.
- Frecuencia de uso de cada pictograma.
- Abandonos.



## 4 Desarrollo

---

En esta sección se explicará los pasos seguidos a la hora de recogida de información, desarrollo de la aplicación, herramientas con las que se ha llevado a cabo este proyecto. También se explicará las partes y ciertos contenidos importantes del proyecto en Android Studio. Para finalizar esta sección se explicarán los principales problemas que se ha tenido mientras se realizaba el proyecto y se hará una detallada explicación del funcionamiento.

### 4.1 Primeros pasos

#### 4.1.1 Búsqueda y recogida de información

El primer objetivo antes de realizar el proyecto fue el estudio del lenguaje Java. Gracias a tutoriales y libros como “How to program Java ninth edition” [3] este estudio se pudo completar. También durante el segundo cuatrimestre de este año, se cursó la asignatura optativa Programación Orientada a Objetos con el fin de asentar los conocimientos en Java estudiados con anterioridad, sirviendo de mucha utilidad.

Una vez tenidos conocimientos suficientes de Java nos centramos en el estudio de Android. Para este estudio nos fue de mucha utilidad el seguimiento de una serie de video tutoriales que se encuentra en la plataforma YouTube<sup>viii</sup>. En esta serie de video tutoriales se realizaron pequeñas aplicaciones para familiarizarse con la herramienta Android Studio.

Tras habernos familiarizado con la herramienta nos centramos en Wear Os. Se continúa investigando y encontramos un curso online “Android Wear Development by Google” ofertada por la plataforma Udacity<sup>ix</sup> que se realiza.

Una vez realizados todos estos pasos previos, se consideró que el nivel era suficiente para empezar a realizar la aplicación con un nivel más que aceptable.

#### 4.1.2 Herramientas utilizadas

Las herramientas utilizadas para realizar el proyecto han sido varias. Como elemento principal donde se ha llevado todo el desarrollo y donde se encontraba el entorno de desarrollo utilizado, Android Studio, ha sido un ordenador personal. Otro elemento utilizado, ha sido un smartphone con sistema operativo Android 6.0.1. En este dispositivo se tuvo que instalar la aplicación Wear Os que se encuentra en la Play Store para poder vincular el dispositivo móvil. Se necesitó el uso de herramientas de diseño gráfico para el retoque de los pictogramas. Y por último se ha utilizado un smartwatch, para este proyecto en específico se ha utilizado un Motorola Moto 360<sup>x</sup>, con el cual se han realizado todas las pruebas, con sistema operativo Android Wear 2.12.0.

## 4.2 Aplicación

### 4.2.1 Nombre y duración del proyecto

La aplicación con nombre **Quiero Decirte** es el resultado del trabajo de un curso entero.

La primera mitad del curso se dedicó como se ha comentado con anterioridad a la formación. Los cursos llevados a cabo, han tenido diferentes duraciones. Por poner un ejemplo, el curso “Android Wear Development by Google” tuvo una duración aproximada de dos semanas y media para entender todos los conceptos explicados y poder ser utilizados posteriormente. La otra mitad del curso, unos 5 ó 6 meses, se dedicó enteramente al desarrollo completo de la aplicación y a la memoria correspondiente.

En estos últimos meses ha habido numerosas versiones donde se han ido haciendo pequeñas modificaciones hasta llegar a la versión definitiva. Gracias a las reuniones periódicas con el tutor se ha conseguido finalizar el proyecto en el tiempo estimado en un primer lugar.

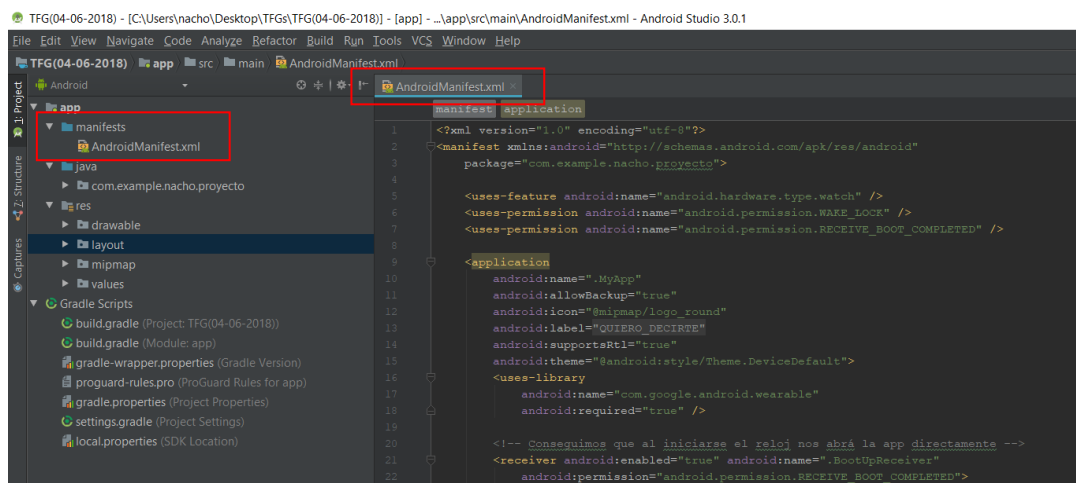
### 4.2.2 Estructura del proyecto en Android Studio

La estructura del proyecto en Android Studio este definido por una serie de carpetas que serán comunes a cualquier otro proyecto ya sea para crear una aplicación Android o para Wear OS, lo que variará es el contenido de cada una de ella. A continuación se explicará cada una de estas carpetas y su contenido.

#### 4.2.2.1 Carpeta Manifest

Esta carpeta es una subcarpeta de la carpeta app que contiene el fichero AndroidManifest.xml.

En este fichero se encuentra información como puede ser el nombre, versión mínima de la API, bibliotecas, nombre del paquete Java. que sirve de identificador.



### 4.1 Manifest



También se encuentran componentes como las actividades y proveedores de contenido que la integran. En nuestro caso en particular en este fichero se encuentra código utilizado para cumplir las especificaciones acordadas.

La primera de las especificaciones con código en este archivo es conseguir que al iniciar el reloj se abra directamente la aplicación.

```
<!-- Conseguimos que al iniciarse el reloj nos abra la app directamente -->
<receiver android:enabled="true" android:name=".BootUpReceiver"
    android:permission="android.permission.RECEIVE_BOOT_COMPLETED">

    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</receiver>
```

## 4.2 Salir app

Y la segunda de las especificaciones es conseguir anular los botones (táctiles) para que no se puede salir de la aplicación.

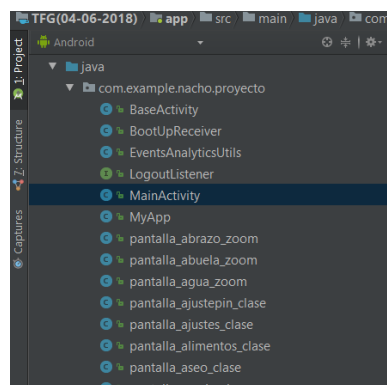
```
<activity
    android:name=".MainActivity"
    android:label="@string/QUIERO_DECIRTE"
    android:theme="@style/Theme.Wearable.Modal"> <!-- Consigo que no se salga de la aplicacion -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

## 4.3 Salir app

### 4.2.2.2 Carpeta Java

Esta carpeta contiene todos los documentos Java que se han realizado en el proyecto. En estos documentos se encuentran todas las funcionalidades requeridas.



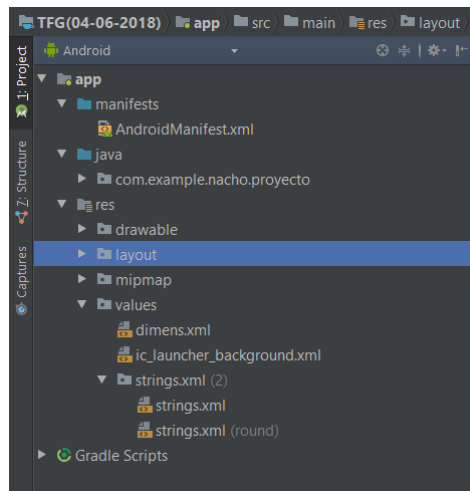
## 4.4 Archivos java

Como se puede observar en la ilustración anterior hay gran cantidad de archivos Java. Esto se debe a la cantidad de pantallas que hay y funciones que se han ido implementando para cumplir con las especificaciones.

#### 4.2.2.3 Carpeta res

La carpeta res o recursos es la carpeta donde se encuentran todos los recursos que se utilizan en los archivos Java. A su vez se encuentra dividida en cuatro subcarpetas:

- Drawable: se encuentran todas las imágenes que se vaya a utilizar en la aplicación.
- Layout.: ficheros donde se controla la estructura visual de cada una de las pantallas de la aplicación.
- Mipmap: se encuentran las imágenes con diferentes densidades de pantalla.
- Values.: carpeta con las variables que serán utilizadas en el código. Se encuentran los recursos que especifican los colores, dimensiones, estilos y cadenas de caracteres.



#### 4.5 Carpeta res

#### 4.2.2.4 Gradle.scripts

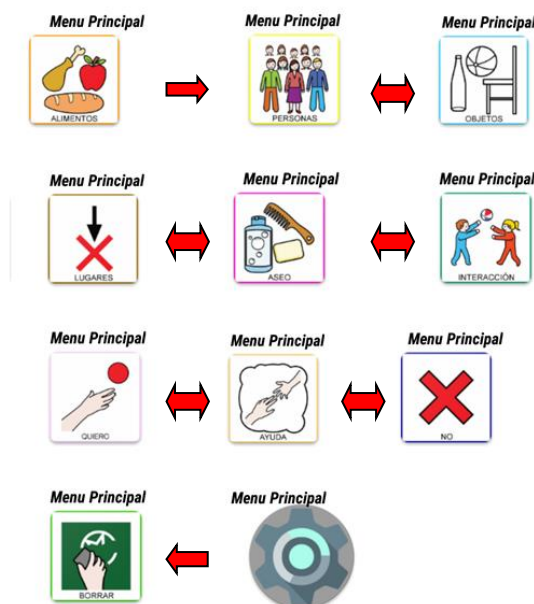
Es el sistema de compilación incorporado en Android Studio.

Con este compilador se genera el código fuente de cada aplicación. Siendo el objetivo generar un fichero empaquetado tipo .apk por medio de los ficheros anteriormente explicados y las bibliotecas correspondientes.

## 4.2.3 Estructura de la Aplicación

### 4.2.3.1 Menú Principal de la aplicación

Es la primera actividad que encontramos nada más inicializar la aplicación. Cuenta con una serie de botones que son las distintas categorías.



### 4.6 Menú Principal

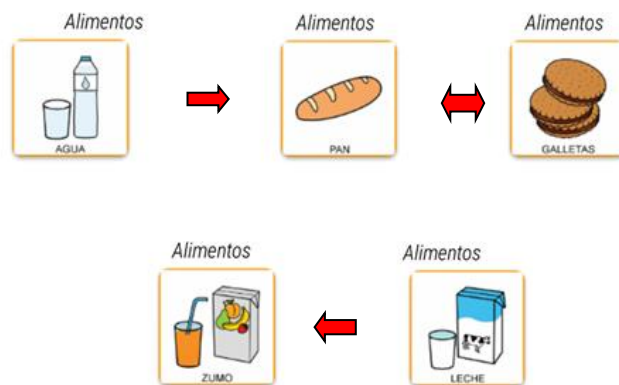
Esta es una representación del menú que se podría ver en el smartwatch. Como no se puede representar en una sola imagen el menú, se ha decidido hacer poco a poco uniéndolas toda al final. Las flechas rojas que se observan representan el tipo de deslizamiento con el dedo para llegar hasta esa opción.

Para desarrollar este menú se ha hecho uso de un HorizontalScrollView como layout. Esto quiere decir que los botones están situados en el eje horizontal.

### 4.2.3.1 Menú Alimentos

Una vez que se selecciona la opción de alimentos en el menú principal, nos llevará a la pantalla de alimentos, [4.7 Pantalla Alimentos](#).

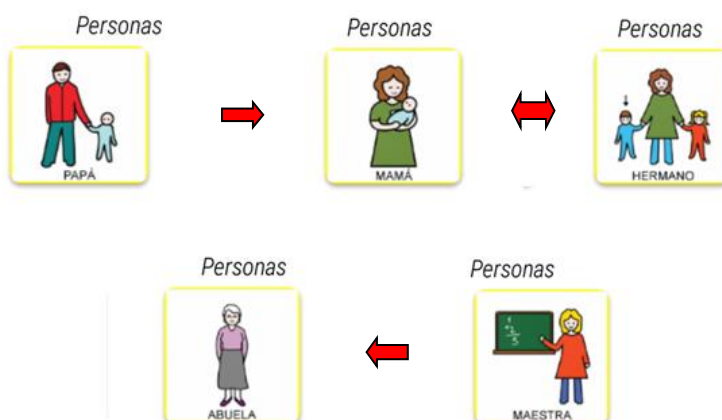
En esta pantalla se pueden ver los 5 elementos que tiene la categoría alimentos. Si se pulsa en cualquiera de los elementos, saldrá una pantalla con el elemento seleccionado en un tamaño mayor.



#### 4.7 Pantalla Alimentos

Para las siguientes categorías se mostrará solo visualmente la estructura ya que cuentan con las mismas características que la pantalla anterior.

##### 4.2.3.2 Menú Personas



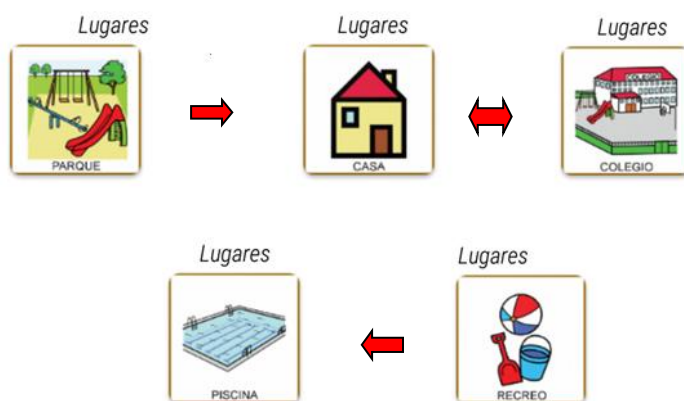
#### 4.8 Pantalla Personas

##### 4.2.3.1 Menú Objetos



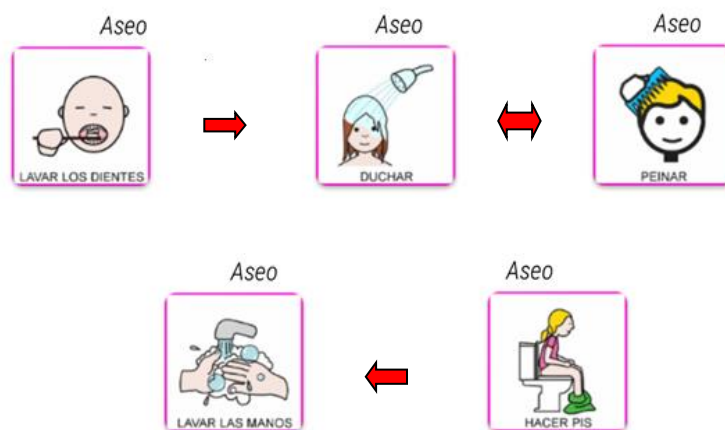
#### 4.9 Pantalla Objetos

#### 4.2.3.2 Menú Lugares



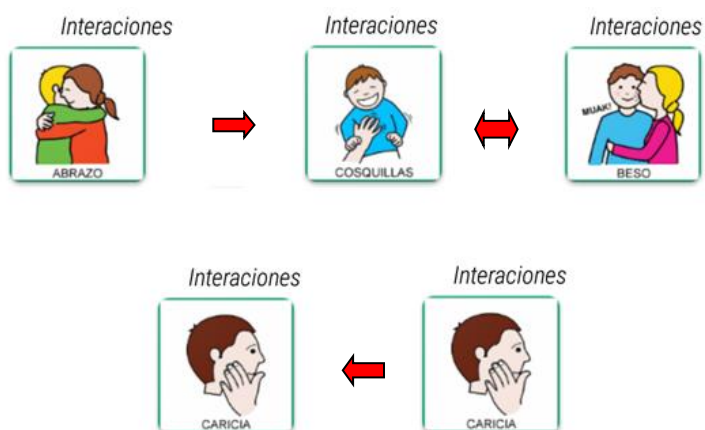
#### 4.10 Pantalla Lugares

#### 4.2.3.3 Menú Aseo



#### 4.11 Pantalla Aseo

#### 4.2.3.4 Menú Interacción



#### 4.12 Pantalla Interacciones

#### 4.2.3.5 Quiero/Ayuda/No/Borrar

Los botones de la pantalla principal Quiero, Ayuda, No y Borrar, no llevan a ningún menú secundario. Al pulsar sobre estos botones únicamente se mostrarán en un mayor tamaño.

Estos botones se encuentran en el menú principal para comunicar con una sola pulsación las acciones más requeridas por los usuarios.

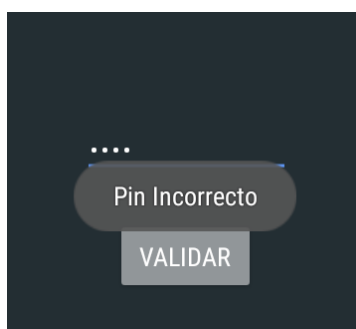
#### 4.2.3.6 Menú Ajustes

Este menú es especial ya que cuenta con dos pantallas. En la primera pantalla se solicita un pin que es conocido por las familias y profesores, nunca por el usuario de la aplicación.



**4.13 Pantalla Pin**

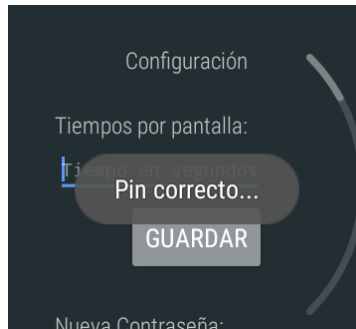
Si el pin es incorrecto, se mostrará un mensaje por pantalla indicándolo.



**4.14 Pin Incorrecto**

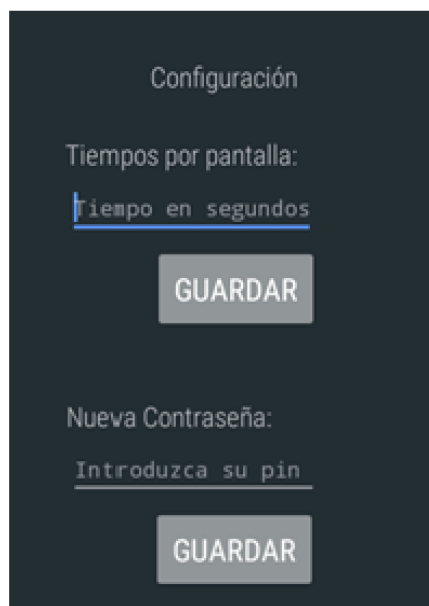
Esta pantalla es muy útil, ya que si un usuario intenta introducir un pin, las probabilidades que acierte son muy bajas, teniendo en cuenta que tendría que acertar tanto la combinación como la cantidad de números del pin solicitado.

En caso de acierto del pin, también se mostrará un mensaje por pantalla y nos llevará a la siguiente pantalla. Esta pantalla no está pensada para que los usuarios lleguen a ella, solo las familias y profesores.



#### 4.15 Pin Correcto

Una vez en la nueva pantalla de configuración, se podrá cambiar el tiempo por pantalla introduciendo un valor en segundos. También se podrá cambiar el pin por defecto, por otro cualquiera.



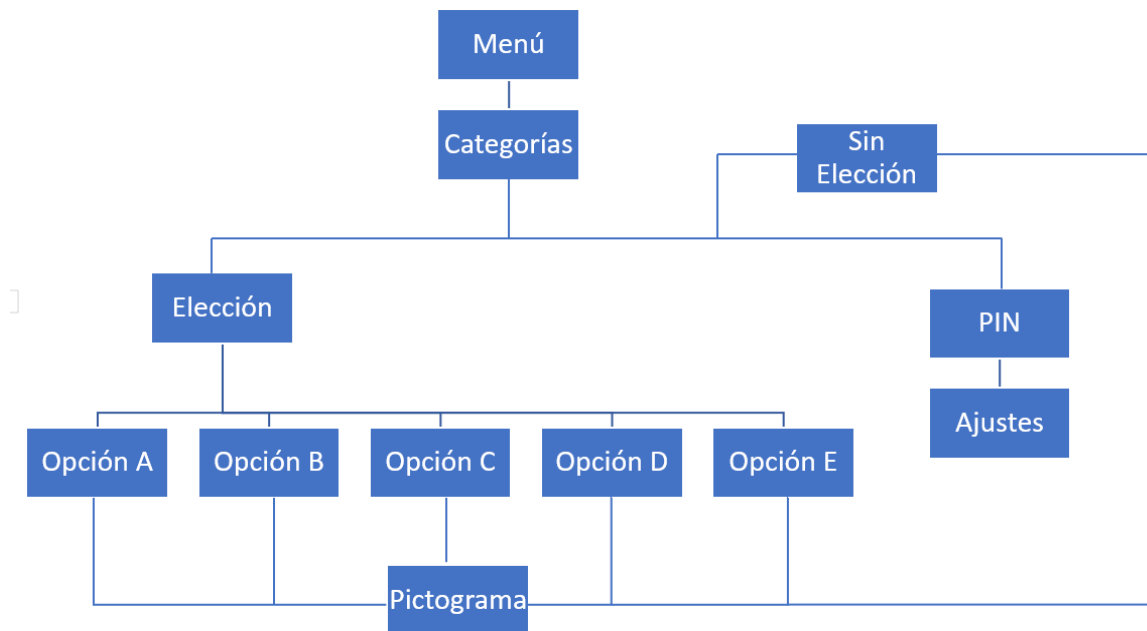
#### 4.16 Pantalla Configuración

Tras introducir el parámetro que se quiere guardar y se pulse el botón correspondiente de guardar, se mostrara un mensaje por pantalla parecido al de las figuras [4.14 Pin Incorrecto](#) y [4.15 Pin Correcto](#).

Para esta pantalla se ha decidido utilizar un `VerticalScrollView` como layout para facilitar y mejorar la visualización de las diferentes opciones.

### 4.3 Estructura final aplicación

La estructura de la aplicación estaba bien definida por las especificaciones desde un principio, por lo que no ha habido casi modificaciones en el transcurso del proyecto.



**4.17 Diagrama Aplicación**

Desde la pantalla inicial en la que hay un menú con categorías, se puede acceder a cada una de ellas. Hay diferentes categorías, se podrían dividir en tres grupos. En el primer grupo, están las categorías que tras seleccionarlás te muestra otro menú con cinco opciones de pictogramas. El segundo grupo, son las categorías que tras seleccionarlás no muestra ningún menú. Estos dos grupos se caracterizan en que ambos terminan mostrando un pictograma en un tamaño considerable. Y en el último grupo, se encontrará el menú de ajustes. En este menú de ajustes, hay una pantalla inicial donde se introduce el pin y si el pin es el correcto se accederá a la configuración.

## ***4.4 Tiempo por pantalla***

El tiempo por pantalla es una de las especificaciones más importantes por lo que le dedicaremos este punto.

Para poder establecer el tiempo máximo por pantalla sin que se esté utilizando, se han tenido que seguir unos pasos hasta conseguirlo.

Primero se ha tenido que crear un cronometro. A su vez ese cronometro tenía que reiniciarse cada vez que el usuario tocara la pantalla.

En segundo lugar, se ha cargado el tiempo por defecto o el tiempo que previamente se ha cambiado en el menú de configuración. Cada vez que el cronometro llegue al tiempo que se ha cargado, la actividad se cerrará y volverá al menú principal.

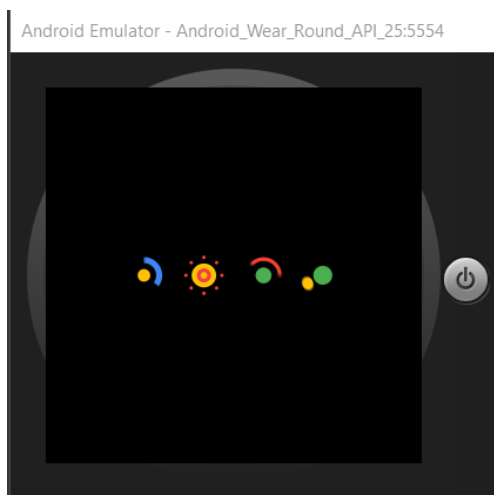
Con estos dos pasos conseguiríamos cumplir con la especificación de tiempo por pantalla, que es realmente útil, si un usuario se bloquea y no sabe que tiene que realizar tras haber llegado a una pantalla determinada.



## 4.5 Principales problemas resueltos

En el transcurso del proyecto han surgido una serie de problemas que se han resuelto de la mejor manera posible. Los principales problemas que se han tenido son:

- Emulador de Android Studio para smartwatch no termina de arrancar quedándose en la pantalla inicial de Android o directamente pantalla en negro completamente.

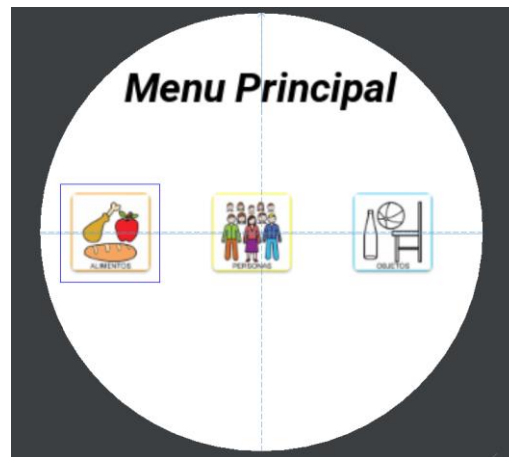


4.18 Emulador

La solución para este problema, tras investigar si era problema del ordenador personal y comunicarlo al tutor, se decide comprar un smartwatch físico.

- Conexión entre el smartwatch y Android Studio. La conexión ha sido uno de los problemas más grandes que se ha tenido, debido a la poca información que hay. Tras horas de investigación y consultas por internet se consigue la deseada conexión explicada en el [Anexo A](#).
- A la hora de mostrar los pictogramas en las diferentes pantallas de la aplicación no se ha podido hacer uso de los diferentes mipmap de los pictogramas.

Esto se debe a que la adaptación de tamaños y densidades de la pantalla no es el óptimo para las especificaciones dadas, ya que Android Studio realiza una adaptación en la que los pictogramas son muy pequeños y no se pueden ver con claridad. Para solucionar este problema se ha tenido que probar cada densidad hasta encontrar la óptima, una vez encontrada se han introducidos todos los pictogramas uno a uno con esa densidad, tarea que ha requerido de mucho tiempo.



#### 4.19 Tamaño mipmap

- Actualización de ciertas funciones de Android. Tras las últimas actualizaciones que realizó Android Studio algunas de las funciones utilizadas tuvieron que ser cambiadas ya que se quedaron obsoletas las que se estaban utilizando.

Una funcionalidad que aunque no se pedía en las especializaciones se implementó, tuvo que ser descartada gracias a las actualizaciones. Esta funcionalidad era muy interesante ya que mandaba un aviso al dispositivo móvil cuando el usuario tocaba los pictogramas así los profesores o la familia sabían exactamente lo que el usuario quería sin necesidad de estar delante en todo momento.

- Integración de la herramienta Firebase. Ciertos problemas con las versiones a la hora integrarlo automáticamente en el proyecto al no ser la última versión disponible.

Esta solución fue más sencilla que las anteriores, lo único que hubo que hacer fue hacer coincidir las versiones de Firebase con la última disponible [4].

## **5 Integración, pruebas y resultados**

---

Una vez que se ha desarrollado la aplicación con todas las especificaciones dadas y llegado a la versión final, nos queda una serie de pasos para la finalización del proyecto.

### **5.1 Publicación**

Una vez que la aplicación haya sido probada por los alumnos, comprobada por los profesores y las familias, y se haya modificado o corregido los posibles errores se procederá a la publicación tanto en Google Play como en la página oficial de la aplicación.

En el [Anexo B](#) se explica de talledamente el proceso que se necesitaría seguir para la publicación.

### **5.2 Resultados**

Los profesores del colegio podrán acceder a través de una cuenta proporcionada por el desarrollador a los resultados. Gracias a las diferentes pantallas de la herramienta Firebase se podrán obtener diferentes resultados, dependiendo lo que el profesor este consultando en ese momento. Los principales resultados que se podrán visualizar en la herramienta son:

- Tiempo medio por pantalla.
- Número de usuarios por mes.
- Número de usuarios por semana.
- Número de usuarios activos diarios.
- Usuarios sin fallos.
- Usuarios que han abandonado antes de llegar a un pictograma final.
- Frecuencia de uso de los pictogramas.

Estos resultados se podrán comparar, descargar o simplemente visualizarlos de manera online. Gracias a estos datos los profesores podrán saber si realmente la aplicación es útil, los pictogramas más utilizados o si por ejemplo el mecanismo hasta llegar al pictograma no se ha entendido correctamente y es necesario cambiar ese mecanismo.



## **6 Conclusiones y trabajo futuro**

---

### **6.1 Conclusiones**

Este Trabajo de Fin de Grado ha tenido como propósito la elaboración de un Comunicador Android sobre reloj inteligente, a través de una aplicación Android. La misma será utilizada en un colegio con alumnos con TEA.

Desde el punto de vista educativo, este TFG ha permitido aprender o consolidar conocimientos como:

- Uso del entorno de desarrollo Android Studio.
- Programación Java.
- Herramientas de desarrollo gráfico.
- Uso de la herramienta Firebase.
- Desarrollar una aplicación Android Wear.
- Publicación de una aplicación.

A esto hay que añadirle, todos los requisitos iniciales y los que se han ido actualizando a medida que ha avanzado el proyecto. Todos estos objetivos han sido bastantes completos para una aplicación de estas características. Por lo que la realización de este Trabajo de Fin de Grado no solo me ha ayudado a mí, ampliando mis conocimientos, sino que también servirá a colegios, familias en la comunicación con los usuarios de esta aplicación.

### **6.2 Trabajo futuro**

El plan de difusión consta de dos etapas entrelazadas. En principio el Comunicador será verificado y optimizado con la colaboración de alumnos de Aulas TEA, Unidades de Educación Especial, Centros de Educación Especial y Asociaciones. En particular:

- Alumnos TEA de la Unidad de Educación Especial del CEIP Romero Peña de La Solana (Ciudad Real)
- Alumnos del Centro Concertado de Educación Especial Autrade (Asociación Regional de Afectados de Autismo y otros Trastornos del Desarrollo) de Ciudad Real.
- Alumnos del Aula TEA del CEIP La Candelaria de Manzanares (Ciudad Real).
- Alumnos TEA del Centro de Educación Especial de Valdepeñas (Ciudad Real).
- Alumnos TEA del Centro de Educación Especial de Tomelloso (Ciudad Real).

Una vez comprobado por estos alumnos, se desarrollará una segunda versión con todas las nuevas indicaciones. La nueva versión se publicará en Google Play y se continuará con las acciones de difusión estándar: presentación en congresos nacionales (CENTAC), publicación en la página web del proyecto donde habrá zona de descargas, manuales de usos, foro para usuarios, links a asociaciones, congresos, aplicaciones similares. También

habrá una página en Facebook con el objetivo de presentar las novedades y llevar tráfico a la página del proyecto para que llegue al máximo número de personas posibles.

Un smartwatch tiene ciertas limitaciones por lo que los trabajos futuros están más limitados que si se estuviera en un dispositivo móvil o tableta. Aun así, hay ciertos trabajos que a día de hoy si se podría implementar.

Esta aplicación está desarrollada para Wear Os por lo que en trabajos futuros podría realizarse una aplicación similar para watchOs (Apple), así tener los dos grandes sistemas operativos de los relojes inteligentes cubiertos.

También, si el equipo de Android realiza nuevas funciones de código que permitan la comunicación de más de un smartwatch con un dispositivo móvil, se podría implementar sobre esta misma aplicación, una función en la que cada vez que un usuario toque el pictograma deseado, se envíe un mensaje a un dispositivo móvil con el usuario correspondiente y el pictograma seleccionado. Con esto se conseguiría que por ejemplo en el colegio, el profesor sepa en todo momento lo que el usuario necesita sin estar delante y poder ir al usuario en específico que lo ha solicitado.

Otra posible mejora para el futuro, podría ser la implementación de un registro de usuarios. Es decir, que cada reloj pueda tener un usuario diferente y que este usuario tenga estadísticas personales a parte de las colectivas que ya tenemos. Gracias a esta mejora, se podría saber si un usuario en específico necesita otro tipo de ayuda o si no comprende el mecanismo de la aplicación. Se podría estudiar a cada usuario individualmente o de forma colectiva.

Otra opción sería la que el profesor o la familia, a través de la pantalla de configuración pueda suprimir, añadir o cambiar los pictogramas que vienen por defecto. En el caso de añadir que las imágenes que se puedan agregar sean de búsquedas de imágenes en Google o del portal ARASAAC.

Aunque el verdadero trabajo futuro es que los colegios donde se va a probar la aplicación, la prueben, como ya se ha indicado antes, y nos indique las posibles modificaciones para poder mejorarla en los siguientes Trabajos de Fin de Grado. Ya que son ellos los que realmente saben las necesidades de los usuarios o como pueden ayudarse de la aplicación para diferentes tareas de aprendizaje o de la vida.

# Referencias

---

- [1] Portal ARAASAC (imágenes): <http://www.arasaac.org/>
- [2] Información Android Studio: <https://developer.android.com/studio/intro/>
- [3] Paul Deitel, Harvey Deitel: “How to program Java (9 edition)” Prentice Hall, 1998
- [4] Firebase (integración y versiones):  
<https://firebase.google.com/docs/android/setup?hl=es-419>
- [5] Google Play Android Developer Console: <https://play.google.com/apps/publish/signup/>
- [6] <https://smartwatchbrands.com/ventas-de-smartwatch-tendencias-2018/>
- [7] <https://www.digitaltrends.com/wearables/wear-os-vs-apple-watch/2/>





## Glosario

---

API	Application Programming Interface
App	Aplicación
DsLab	Digital System Lab
EPS	Escuela Politécnica Superior
UAM	Universidad Autónoma de Madrid
SO	Sistema Operativo
APK	Aplication Package File



## Anexos

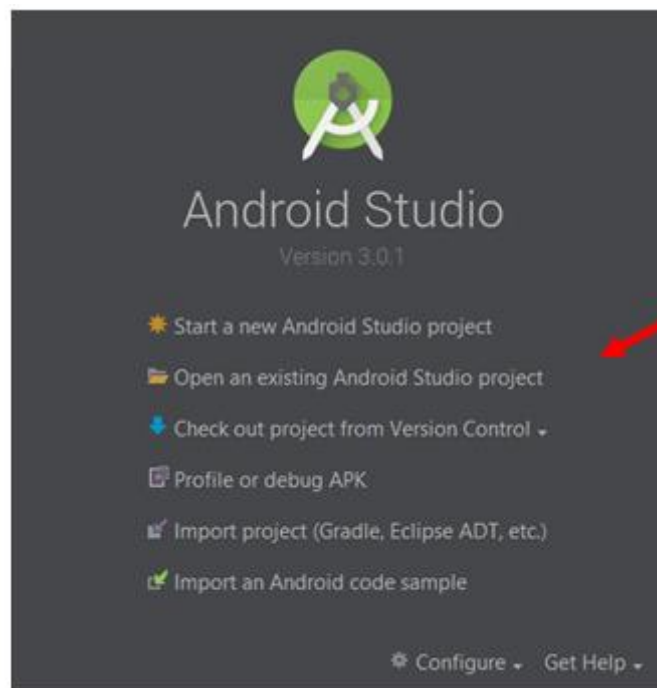
---

### ***A Conexión del smartwatch con Android Studio***

En este anexo se explicará detalladamente como conectar un smartwatch con la herramienta Android Studio. Una vez que el smartwatch esté vinculado a un dispositivo móvil a partir de la aplicación Wear Os.

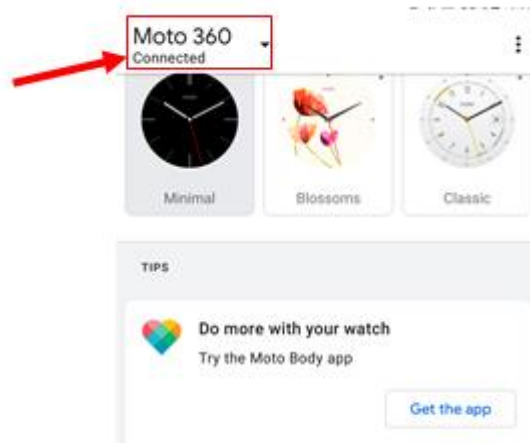
Se realizarán una serie de pasos con el fin de conseguir el propósito:

1. En primer lugar se abre el proyecto previamente creado:



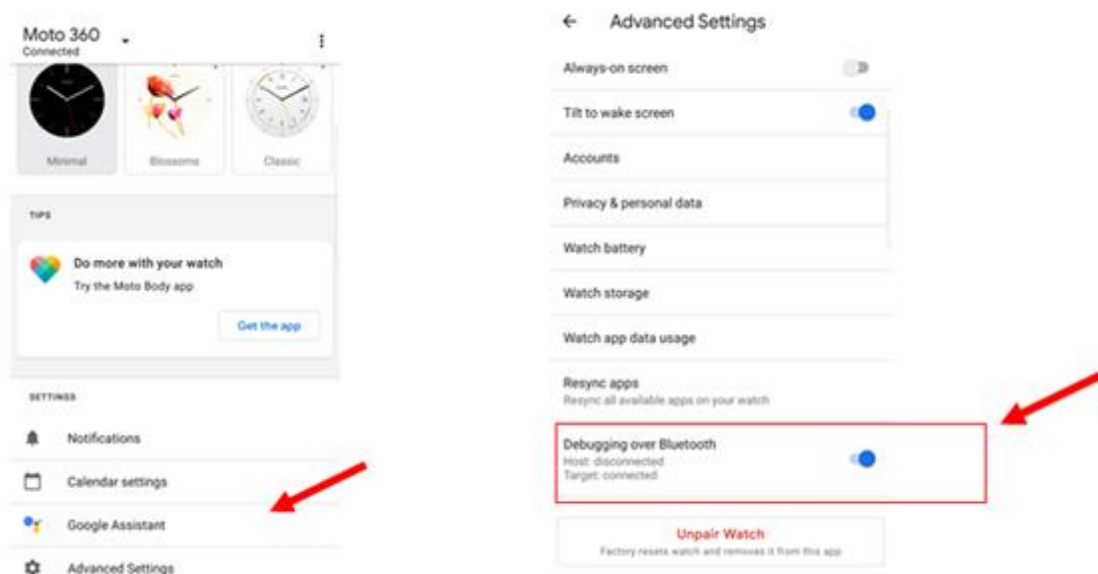
**Figura 0.1 Abrir Proyecto**

2. Se conecta el dispositivo móvil al que se tiene vinculado el smartwatch con un cable USB al ordenador.
3. Nos aseguramos de que el smartwatch conectado correctamente con el dispositivo móvil.



**Figura 0.2 Comprobación**

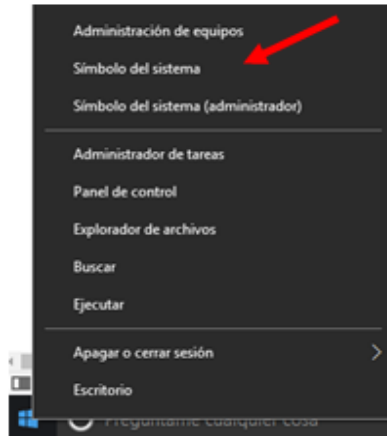
4. Una vez conectado no dirigimos a Advanced Setting (ajustes avanzados) y marcamos la opción de debugging over Bluetooth.



**Figura 0.3 Acceso ajustes**

Una vez que seleccionamos la opción de debugging over Bluetooth, se conectara automáticamente el Target como aparece en la imagen. Nos tocará ahora conecta el host que se encuentra desconectado.

5. En el ordenador necesitaremos utilizar la ventana de comando en nuestro caso la de Windows e introducir una serie de comandos. Para ello buscaremos la aplicación símbolos del sistema.



**Figura 0.4 Símbolos del sistema**

6. Los comandos a introducir son los siguientes:

```
cd ../../..
cd Users
cd Usuario
cd AppData
cd Local
cd Android
cd Sdk
cd platform-tools
adb devices
adb forward tcp:4444 localabstract:/adb-hub
adb connect localhost:4444
adb connect 127.0.0.1:4444
```

Tener especial cuidado  
poniendo el usuario  
correspondiente

**Figura 0.5 Comandos**

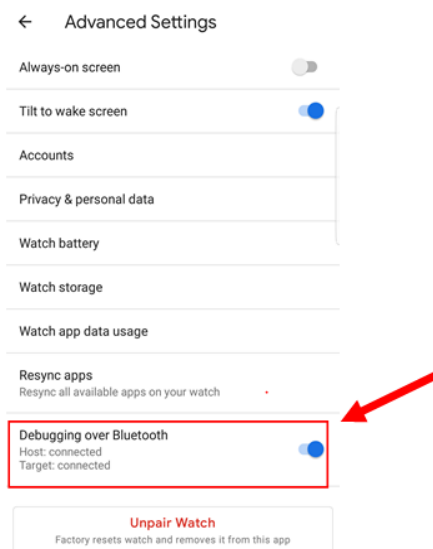
7. Tras introducir todos los comandos y si no ha habido ningún error, ya estará conectado.

```
C:\Users\nacho\AppData\Local\Android\Sdk>cd platform-tools
C:\Users\nacho\AppData\Local\Android\Sdk\platform-tools>adb devices
List of devices attached
127.0.0.1:4444 device
185ebaf device

C:\Users\nacho\AppData\Local\Android\Sdk\platform-tools>adb forward tcp:4444 localabstract:/adb-hub
error: more than one device/emulator

C:\Users\nacho\AppData\Local\Android\Sdk\platform-tools>adb connect 127.0.0.1:4444
already connected to 127.0.0.1:4444
C:\Users\nacho\AppData\Local\Android\Sdk\platform-tools>
```

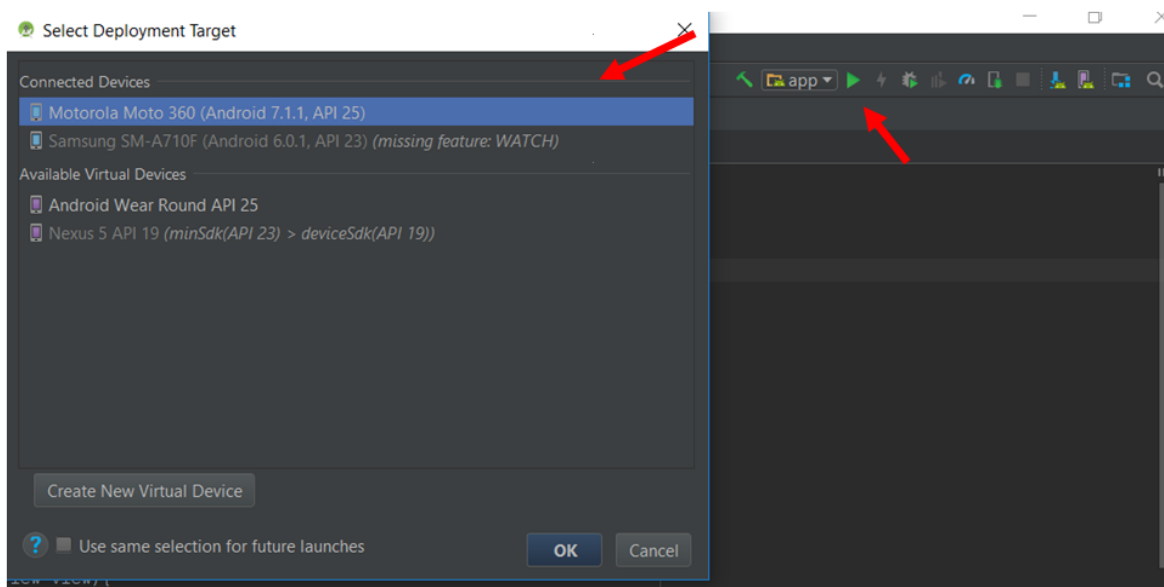
**Figura 0.6 Conexión Ventana**



**Figura 0.7 Conexión Wear Os app**

Como se puede ver en ambas ilustraciones, nuestro smartwatch ya estaría conectado y listo para usar.

8. En Android Studio solo nos falta darle al play como aparece en la figura y en nuestro reloj correspondiente para poder probar la aplicación desarrollada.

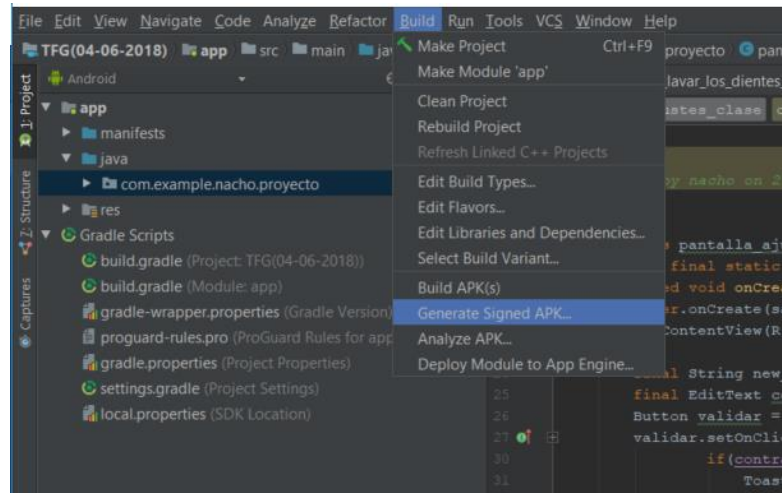


**Figura 0.8 Conectar último paso**

## B Publicación en Google Play

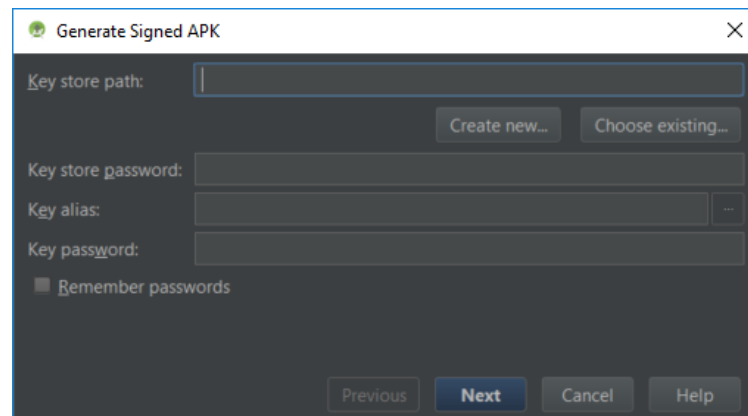
Para la publicación de una aplicación es necesario seguir una serie de pasos:

- 1) Generar el archivo apk, este archivo es necesario para la publicación ya que es el que se subirá a Google Play.



**Figura 0.9 Build**

En el menú de la parte superior seleccionamos primero Build y en el menú desplegable Generate Signed APK... como aparece en la figura anterior. Nos aparecerá una pantalla como la siguiente:



**Figura 0.10 Generate Signed APK**

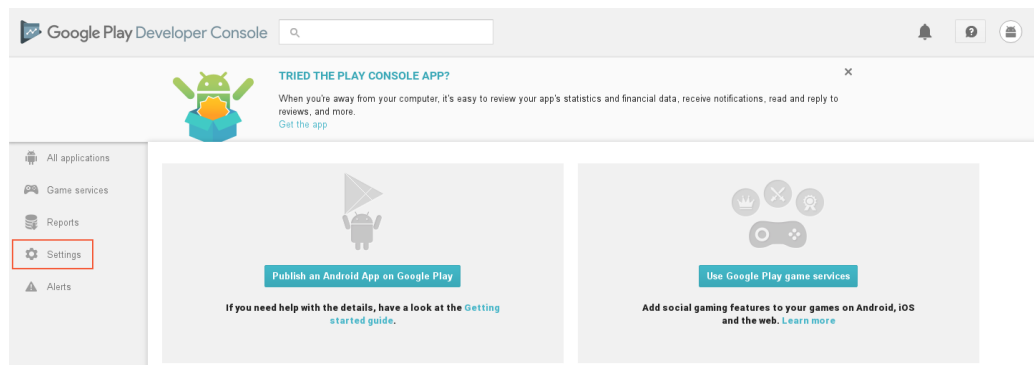
Cuando estemos en esta pantalla seleccionamos Create new para crear una nueva clave. Ya que con la clave que estábamos trabajando solo sirve para hacer prueba y no para la distribución. Rellenamos los distintos campos que nos aparece en la nueva pantalla, teniendo especial cuidado con la clave. Esta clave tendremos que guardarla ya que es muy importante para pasos siguientes.

Una vez que rellenemos los campos solo tendremos que dar a Next y nos generará un archivo con extensión .apk.

- 2) Accederemos a la página web “Google Play Android Developer Console” [5] con la cuenta del usuario con la que queremos subir la aplicación. En nuestro caso, la cuentas del laboratorio DSLab.

Si es la primera vez que realizas esta tarea, te pedirá aceptar el acuerdo para desarrolladores, pagar la cuota de registro y rellenar la información de tu cuenta. En nuestro caso, no es la primera vez por lo que estos pasos no habrá que realizarlos.

- 3) Una vez dentro de la cuenta, en el panel seleccionamos Settings y Publish an Android App on Google Play.



**Figura 0.11 Settings<sup>xi</sup>**

Nos lo primero que nos pedirá es seleccionar el lenguaje y el nombre de la aplicación. Nada más rellenar estos dos campos tendremos que subir el archivo apk generado con anterioridad. Cuando hayamos subido el archivo nos pedirá rellenar una serie de datos:

- Dos descripciones de la aplicación, una más de tallada que la otra..
- Diferentes capturas de pantalla, para que se puedan ver en la página de Google Play.
- El icono de la aplicación en diferentes tamaños y densidades.
- Categoría en la que queremos introducir la aplicación.
- Precio.
- Política de privacidad (solo si es necesario).
- Distribución (países).

Tras haber rellenado todos estos campos seleccionaremos la opción de publicar aplicación que está situada arriba a la derecha.





**Figura 0.12 Publicar aplicación**

Google Play revisará si todo se ha hecho correctamente y mandará un correo informando cuando esta publicada la aplicación.

**Nota:**

Antes de generar el archivo .apk se deberá cambiar en Firebase la clave SHA 1 que estaba (prueba), por la nueva clave SHA1 de producción. También se deberá descargar el fichero google-services.json e introducirlo en la carpeta del proyecto, como ya se había hecho en la integración.



## **C Referencias Imágenes**

- 
- i <http://www.fundacionorange.es/aplicaciones/dictapicto-tea/>
  - ii <http://www.pictogramas.es/PictogramasAPP.html>
  - iii <http://www.fundacionorange.es/aplicaciones/emoplay/>
  - iv <http://www.graceapp.com/>
  - v <https://smartwatchbrands.com/ventas-de-smartwatch-tendencias-2018/>
  - vi <https://www.idc.com/getdoc.jsp?containerId=prUS43598218>
  - vii <https://elandroidelibre.lespansol.com/2018/03/android-wear-ahora-wear-os-cambio-mucho-sentido.html>
  - viii <https://www.youtube.com/playlist?list=PLU8oAlHdN5Bkn-KS1sRFISEnXXcAtAJ9P>
  - ix <https://eu.udacity.com/course/android-wear-development--ud875A>
  - x <https://www.motorola.es/products/moto-360>
  - xi <https://docs.microsoft.com/es-es/xamarin/android/deploy-test/publishing/publishing-to-google-play/?tabs=vswin>